

An Algorithm for System Identification of a Discrete-Time Polynomial System without Inputs – Extended Version [★]

Jana Němcová, ^{*} Mihály Petreczky, ^{**} Jan H. van Schuppen ^{***}

^{*} *Department of Mathematics, Institute of Chemical Technology,
Prague, Czech Republic (e-mail: Jana.Nemcova@vscht.cz)*

^{**} *Dept. of Comp. Science & Automatic Control, Écoles des Mines de Douai,
Douai, France (e-mail: mihaly.petreczky@mines-douai.fr)*

^{***} *Van Schuppen Control Research, Amsterdam, The Netherlands
(e-mail: jan.h.van.schuppen@xs4all.nl)*

Abstract: A subalgebraic approximation algorithm is proposed to estimate from a set of time series the parameters of the observer representation of a discrete-time polynomial system without inputs which can generate an approximation of the observed time series. A major step of the algorithm is to construct a set of generators for the polynomial function from the past outputs to the future outputs. For this singular value decompositions and polynomial factorizations are used. An example is provided.

Keywords: System identification, algorithms, nonlinear systems, algebraic systems theory.

1. INTRODUCTION

The system identification of polynomial systems is motivated by the need for models of biochemical reaction systems in the life sciences. Also in the area of control engineering and of economics there is a need to determine parameter values of such control systems from data. As far as the authors have been able to determine there is no satisfactory algorithm for the general problem of determining the parameter values of these systems.

The problem of the paper is to determine a system in the class of discrete-time polynomial systems without inputs in the form of an observer realization such that it produces for each time series of outputs a predicted time series which is a reasonable approximation of the supplied output time series.

The relevant literature on polynomial systems and their system identification is briefly summarized. At the time this paper is written there are available results on the realization theory of polynomial and of rational systems see Sontag (1979), Bartosiewicz (1988), Němcová and van Schuppen (2009), Němcová and van Schuppen (2010). The problem of structural identifiability of polynomial and of rational systems was solved by J. Němcová in Němcová (2010). Earlier papers of the authors include Nemcova and van Schuppen (2009), Němcová et al. (2012). Various aspects of system identification of rational systems are also discussed in Boulier and Lemaire (2009), Gevers et al. (2013), Bazanella et al. (2014).

The algorithm proposed in this paper determines a polynomial system in the form of an observer, thus driven by the available output. It will be proven using system theory that an observer polynomial system is equivalent with the conditions: (1) the state of the observer at any time is a polynomial function of the past outputs; (2) the future outputs are a polynomial

function only of the current state; and (3) the next state is a polynomial function only of the current state and the current output. The main step of the algorithm is to construct an approximate generator set for the polynomial equation from the past outputs to the future outputs. The subalgebra generated by this generator set is then an approximate subalgebra of the algebra of the function from past outputs to future outputs.

2. PROBLEM FORMULATION

System identification of a polynomial and rational system is motivated by the occurrence of these systems in engineering (satellite orientation problems), biochemical reaction networks (mass action kinetics), and economics (products of prices and quantities). In this paper the authors focus on polynomial systems. For the extension to rational systems and to systems with inputs there is insufficient space in this short paper.

System identification is a research area that addresses the problem of how to go from observational data to a system with its parameter values. The following procedure is often used: (1) *Modeling*. Model the phenomenon as a control system as understood in system theory; (2) *Data collection*. Collect data in the form of a time series of the phenomenon to be identified; (3) *Structural identifiability*. Determine whether the selected parametrization of the subclass of systems is structurally identifiable and, if not, modify the system subclass such that the system subclass is structurally identifiable; (4) *Approximation*. Determine an algorithm to compute the parameters of a system in the considered subclass from the observation data which is a reasonable approximation according to an approximation criterion; (5) *Complexity estimation*. Determine that subclass of systems which achieves a reasonable value for the approximation criterion and which minimizes the complexity. This paper addresses Step (4).

For the approximation problem of system identification there are two main approaches: (1) Minimization of an approxima-

[★] Jana Němcová thanks the national agency of the Czech Republic for financial support by GAČR Project 13-16764P.

tion criterion over the considered subclass of systems. This approach, though often used, suffers from the problem that the criterion is a nonconvex function of the parameters which makes the minimization approach practically almost impossible. (2) An algebraic method based on realization theory of system theory. The method is explained in the next section.

For the remainder of this paper the reader is expected to have read the notation and terminology of the appendices.

Definition 2.1. A *time series* is a collection of real vectors denoted by its dimension $d_y \in \mathbb{Z}_+$, its length in time steps $t_1 \in \mathbb{Z}_+$, and $\{\bar{y}(t) \in \mathbb{R}^{d_y}, t \in \{1, 2, \dots, t_1\} \subset \mathbb{Z}\}$. Often there is available a finite set of time series.

Definition 2.2. The system class considered is that of a discrete-time polynomial system without inputs in observer representation, which produces $y(t|t-1)$, a prediction of the next output based on past outputs.

$$\begin{aligned} x(t+1) &= f_o(x(t), y(t)), \quad x(0) = x_0, \\ y(t|t-1) &= h_o(x(t)), \\ T &= \{0, 1, 2, \dots, t_1\} \subset \mathbb{N}, \quad X = \mathbb{R}^n, \quad Y = \mathbb{R}^{d_y}, \\ f_o : X \times Y &\rightarrow X, \quad h_o : X \rightarrow Y, \end{aligned}$$

where f_o and h_o are both polynomial functions.

Problem 2.3. Consider a finite set of time series and the subclass of control systems of Def. 2.2. The problem is to determine an observer system in the subclass considered, specified by its parameter values, such that the estimated observer system, when supplied with the output of the time series, produces a one-step prediction of the output which prediction time series is close to the observed time series.

3. THE APPROACH

The subalgebraic approach to system identification of polynomial systems is based on the subspace identification algorithm of Gaussian systems. The approach was initially proposed by H. Akaike based on contacts with R.E. Kalman, has been developed for Gaussian systems, and is known as the subspace identification algorithm, see for references van Overschee and De Moor (1996). The subalgebraic approach to polynomial systems is based on realization theory of nonlinear systems in particular of bilinear, of polynomial and of rational systems. See for references on realization theory those mentioned in the previous section and Fliess (1990), Isidori (1973), Vidal (2008). No approximation criterion is used in the paper. The principle of the subalgebraic method is explained with the following theorem.

Theorem 3.1. Consider a discrete-time polynomial system without inputs in its observer realization with as output the one-step prediction,

$$x_o(t+1) = f_o(x_o(t), y(t)), \quad x_o(0) = x_{o,0}, \quad (1)$$

$$y(t|t-1) = h_o(x_o(t)), \quad f_o, \quad h_o, \quad \text{polynomial maps.} \quad (2)$$

The observer system representation (1,2) may be transformed to the following set of polynomial functions assuming that the observer is a true observer, hence the predictions equal $y(t) = y(t|t-1)$, and in terms of $(y^+(t), y^-(t))$ as defined in equation (8),

$$x_o(t) = g_{io}(x_{o,0}, y^-(t)), \quad (3)$$

$$y^+(t) = h_{io}(x_o(t)) = h_{io}(g_{io}(x_{o,0}, y^-(t))), \quad (4)$$

$$x_o(t+1) = f_o(x_o(t), y(t)), \quad f_o, \quad g_{io}, \quad h_{io}, \quad \text{polyn.} \quad (5)$$

Note that by the equations (3,4) the current state is a polynomial of the components of the past outputs and the future outputs are polynomial functions of the components of the current state; and by equation (5) the next state $x_o(t+1)$ is a polynomial map in the tuple $(x_o(t), y(t))$ of the current state and the current output. If one considers the initial state as a constant then the equations (3,4) define for any time a polynomial function from the past outputs to the future outputs.

The algorithm for the subalgebraic approximation is based on the above theorem and consists of the steps:

- (1) Compute a state vector $x(t)$ in terms of a polynomial function of past outputs such that the future outputs are a polynomial function of it. In terms of formulas,

$$y^+(t) \approx f_{io}(y^-(t)) = h_{io}(g_{io}(y^-(t))) = h_{io}(x(t)), \quad (6)$$

$$x(t) = g_{io}(y^-(t)), \quad f_{io}, \quad g_{io}, \quad h_{io} \text{ polynomial functions.}$$

- (2) Compute the polynomial system dynamics

$$x(t+1) \approx f_o(x(t), y(t)). \quad (7)$$

The main task of the algorithm is to compute a set of generators of the polynomial map f_{io} from past outputs to future outputs. The complexity of the computations is limited by several steps of the algorithm.

4. THE ALGORITHM

Definition 4.1. The *subalgebraic approximation algorithm* for system identification of discrete-time polynomial systems. Data: A time series of outputs with the notations: the dimension of the output $d_y \in \mathbb{Z}_+$, the length of the time series $t_1 \in \mathbb{Z}_+$, the number of time series $s \in \mathbb{Z}_+$, and finally the time series matrix $Y_{ts} \in \mathbb{R}^{t_1 \times d_y \times s}$. The parameters of the algorithm are $r_1, r_2, r_3, r_4 \in (0, 1) \subset \mathbb{R}$ and (t_{\min}^+, t_{\min}^-) , $(t_{\max}^+, t_{\max}^-) \in \mathbb{Z}_+^2$ defined below, and various maximal power vectors.

- (1) *Construct the vectors of the past and of the future time series.* Take a time $t \in T = \{1, 2, \dots, t_1\}$ less or equal to $t_1/2$. Denote the length of the tuple of the future and of the past output time series respectively by $(t^+, t^-) \in \mathbb{Z}_+$ and set their extrema such that $(t + t_{\max}^+), (t - t_{\max}^-) \in T$.

Iterate from Step 2 to Step 6 in a Levinson-like manner by increasing the horizon lengths (t^+, t^-) from the values (t_{\min}^+, t_{\min}^-) to the values $(t_{\max}^+, t_{\max}^-) \in \mathbb{Z}_+^2$.

Construct the symbolic vectors of the future and the past series, and their values for the each of the time series,

$$\begin{aligned} d_{y^+} &= t^+ d_y, \quad d_{y^-} = t^- d_y \in \mathbb{Z}_+, \\ (y^+(t), y^-(t)) &\in (\mathbb{R}^{d_{y^+}} \times \mathbb{R}^{d_{y^-}}) \end{aligned} \quad (8)$$

$$= \left(\begin{pmatrix} y(t + t^+ - 1) \\ y(t + t^+ - 2) \\ \vdots \\ y(t) \end{pmatrix}, \begin{pmatrix} y(t - 1) \\ y(t - 2) \\ \vdots \\ y(t - t^-) \end{pmatrix} \right).$$

$$(y^+(t, k), y^-(t, k)) \in (\mathbb{R}^{d_{y^+}} \times \mathbb{R}^{d_{y^-}}), \quad k \in \mathbb{Z}_s.$$

- (2) *Define the power matrices of the future and past output time series.*

$$d_{v+} = d_{y+}, K_{v+} = I_{d_{v+}} \in \mathbb{N}^{d_{v+} \times d_{y+}},$$

$$k_{\max,y} \in \mathbb{N}^{d_y}, k_y^* = \max_{i \in \mathbb{Z}_{d_y}} k_{\max,y}(i), \quad (9)$$

$$k_{\max,y-} = (k_{\max,y}^T, \dots, k_{\max,y}^T)^T \in \mathbb{N}^{d_{y-}},$$

$$d_{v-} \leq \left(\prod_{i=1}^{d_y} (k_{\max,y}(i) + 1) \right)^{-} \leq (k_y^* + 1)^{d_{y-}},$$

$$K_{v-} \in \mathbb{N}^{d_{v-} \times d_{y-}}(k_{\max,y-}),$$

$$(L_{v+}, K_{v+}) = (I_{d_{v+}}, I_{d_{v+}}),$$

$$(L_{v-}, K_{v-}) = (I_{d_{v-}}, K_{v-}).$$

- (3) *Iteration of blocks of the power matrix K_{v-} .* If the row dimension of the bounded power matrix K_{v-} of the past outputs is very high, say larger than 500, then execute this step. Partition the full power matrix K_{v-} into a finite number of row blocks $K_{v-}^{(1)}, \dots, K_{v-}^{(m_1)}$ starting with lowest power blocks. Denote the corresponding row dimensions by $d_{v-}^{(1)}, \dots, d_{v-}^{(m_1)}$.

Iterate from Step 4 to Step 5. Start with the first block $(d_{v-}^{(1)}, L_{v-}^{(1)}, K_{v-}^{(1)})$. After each cycle add to the current generator set indexed by $(L_{v-}^{(g)}, K_{v-}^{(g)})$ the next block of the power matrix.

- (4) *Construct the monomial vectors of the future and the past outputs.* Construct next for the parameters $(d_{v+}, K_{v+}, d_{v-}^{(m)}, K_{v-}^{(m)})$ set in Step 2 or Step 3, the associated monomial vectors according to Def. B.2,

$$v^+(k) = v(y^+(t, k), d_{y+}, K_{v+}) = y^+(k) \in \mathbb{R}^{d_{v+}} \quad (10)$$

$$V^+(t) = (v^+(1) \ v^+(2) \ \dots \ v^+(s)) \in \mathbb{R}^{d_{v+} \times s}, \quad (11)$$

$$v^-(k) = v(y^-(t, k), d_{y-}, K_{v-}^{(m)}) \in \mathbb{R}^{d_{v-}^{(m)}}, \quad (12)$$

$$v_i^-(k) = \prod_{j=1}^{d_{y-}} y_j^-(t, k) K_{v-}^{(m)}(i, j), \text{ see (B.1),} \quad (13)$$

$$V^-(t) = (v^-(1) \ v^-(2) \ \dots \ v^-(s)) \in \mathbb{R}^{d_{v-}^{(m)} \times s}. \quad (14)$$

Note that the dimension and hence the complexity of v^- is exponential in terms of $d_{v-} = t^- d_y$.

- (5) *Reduce the generator set by (1) linear dependence.* Compute, according to Algorithm E.4, the approximate monomial equation of the future and the past time series.

$$(n_1, D_{n_1}, C_{v+}, L_1, X, H^*, \text{table}_1) \quad (15)$$

$$= \text{SVDtrunc}(dy^+, dy^-, dv_{y+}, dv_{y-}, s,$$

$$V^+(t), V^-(t), r_1); H^* \in \mathbb{R}^{d_{v+} \times d_{v-}},$$

$$V^+(t) \approx H^*(t) V^-(t) = C_{v+} L_1 (y^-(t, *))^{K_{v-}^{(m)}}, \quad (16)$$

$$C_{v+} \in \mathbb{R}^{d_{v+} \times n_1}, L_1 \in \mathbb{R}^{n_1 \times d_{v-}^{(m)}}. \quad (17)$$

- (6) *LK-Reduction.* Reduce the generator set further for the matrix tuple $(L_1, K_{v-}^{(i)})$ by deleting those columns of the matrix L_1 and the corresponding rows of the matrix K_1 whose l_1 -norm of the column is less than $r_4 \in (0, 1)$ times the l_1 -norm of L_1 . Thus delete column j of L_1 if,

$$\sum_{i=1}^n |L_1(i, j)| \leq r * \|L_1\|_{L_1} = r * \max_{j \in \mathbb{Z}_n} \sum_{i=1}^n |L_1(i, j)|,$$

$$(L_{v-}^{(g)}, K_{v-}^{(g)}), \text{ result.} \quad (18)$$

- (7) *Reduce the generator set by (2) elimination of products of generators.* Compute a new generator set with possibly less generators according to Step 2 of the algorithm of Def. E.2. Starting from equation (18). with $(C_{v+}, (L_{v-}^{(g)}, K_{v-}^{(g)}))$. The result is,

$$y^+(t) = v^+(t) \approx C_{v+} L_{v-}^{(g)} (y^-(t))^{K_{v-}^{(g)}}$$

$$\approx h_{io}^+(x(t)) = L_{h_{io}^+} x(t)^{K_{h_{io}^+}}, \quad (\text{B.1}) \quad (19)$$

$$x(t) = g_{io}(v^-(t)) = L_{g_{io}} y^-(t)^{K_{g_{io}}} \in \mathbb{R}^n. \quad (20)$$

- (8) *Compute the output equation.*

$$y(t) = P_{y(t)} y^+(t) \approx P_{y(t)} h_{io}^+(x(t)) = h_o(x(t)),$$

$$P_{y(t)} \in \mathbb{R}^{d_y \times d_{y+}}, \text{ a projection.}$$

The next steps aim at the computation of the system dynamics, see equation (7).

- (9) *Compute the value of the next state.* First compute the past time series at the next time index $(t + 1)$. Secondly, compute the value of $X(t + 1)$ for each time series.

$$y^-(t + 1, j) = \begin{pmatrix} y(t, j) \\ y(t - 1, j) \\ \vdots \\ y(t - t^- + 1, j) \end{pmatrix} \in \mathbb{R}^{d_{y-}}, \quad (21)$$

$$v^-(t + 1, j) = v(y^-(t + 1, j), d_{y-}, K_{g_{io}}) \in \mathbb{R}^{d_{v-}}, \quad (22)$$

$$V^-(t + 1) = (v^-(t + 1, 1) \ v^-(t + 1, 2) \ \dots \ v^-(t + 1, s)) \quad (23)$$

$$\in \mathbb{R}^{d_{v-} \times s}; d_{v_x} = n \in \mathbb{Z}_+,$$

$$V_x(t + 1) = X(t + 1) = L_{g_{io}} V^-(t + 1) \in \mathbb{R}^{n \times s}.$$

- (10) *Compute the monomial vector of the current state and the current output.*

$$d_{(x,y)} = n + d_y, d_{v_{(x,y)}} \in \mathbb{Z}_+, \quad (24)$$

$$d_{v_{(x,y)}} = \prod_{i=1}^n [k_{\max,x}(i) + 1] \prod_{j=1}^{d_y} [k_{\max,y2}(j) + 1], \quad (25)$$

$$\text{choose } k_{\max,x} \in \mathbb{N}^n, k_{\max,y2} \in \mathbb{N}^{d_y},$$

$$k_{\max,(x,y)} = (k_{\max,x}^T k_{\max,y2}^T)^T \in \mathbb{N}^{d_{(x,y)}},$$

$$K_{v_{(x,y)}} \in \mathbb{N}^{d_{v_{(x,y)}} \times d_{(x,y)}}(k_{\max,(x,y)}),$$

$$v_{(x(t), y(t))}(k) = v\left(\begin{pmatrix} x(t, k) \\ y(t, k) \end{pmatrix}, (n + d_y), K_{v_{(x,y)}}\right) \quad (26)$$

$$V_{x,y} = (v_{(x(t), y(t))}(1) \ \dots \ v_{(x(t), y(t))}(s)). \quad (27)$$

Note that the complexity of the expression of $d_{v_{x,y}}$ is exponential in $n + d_y$.

- (11) *Reduce the generator set by (1) linear dependence.* Compute the approximate polynomial function of the next future state depending on the vector of the current state and of the current output. Compute according to Def. E.4,

$$(n_2, D_{n_2}, C_2, L_2, X_2, H_2^*, \text{table}_2) \quad (28)$$

$$= \text{SVDtrunc}(n, d_{(x,y)}, n, dv_{(x,y)}, s, V_x(t + 1),$$

$$V_{x,y}, r_2); L_3 = H_2^* \in \mathbb{R}^{n \times d_{v_{(x,y)}}}.$$

$$X(t + 1) = V_{x(t+1)} \approx L_3 V_{(x(t), y(t))}, \quad (29)$$

$$x(t + 1) = L_3(x(t), y(t))^{K_3}. \quad (30)$$

- (12) *Approximate the monomial map.* Reduce the matrix tuple (L_3, K_3) to $(L_{f_{io}}, K_{f_{io}})$ as in Step 6 for the matrix pair $(L_2, K_{v-}^{(i)})$.

(13) *Compute the polynomial system.* Finally compute the discrete-time polynomial system without input in the observer form, by writing the linear map of nomomials as a vectorial polynomial function,

$$f_o(x, y) = L_{f_{io}}(x, y)^{K_{f_{io}}}, \text{ see (B.1),} \quad (31)$$

$$x_o(t+1) = f_o(x_o(t), y(t)), \quad x_o(0) = x_{o,0}, \quad (32)$$

$$y_o(t|t-1) = h_o(x_o(t)), \text{ from Step (8),} \quad (33)$$

$$X_{o,0} = X(t) \in \mathbb{R}^{n \times s}. \quad (34)$$

Output $(n, f_o, h_o, \{X_{o,0}(k), y^+(k), y^-(k), k \in \mathbb{Z}_s\})$.

The user is advised to take t^+ and t^- equal to about 4 times the dimension of the expected state space.

5. EXAMPLES

Example 5.1. For a set of times series generated by a simple polynomial system with a one-dimensional output and a two-dimensional state vector, a computer program for Algorithm 4.1 has computed the following observer polynomial system.

$$\begin{aligned} x(t+1) &= f_o(x(t), y(t)) = L_{f_o}(x_o(t), y(t))^{K_{f_o}}, \\ y(t|t-1) &= h_o(x(t)) = Cx_o(t), \quad C = (-0.0225 \ 0.0336), \\ L_{f_o} &= \begin{pmatrix} 0.009 \ .089 & 0.023 \ .571 & -.004 & -.020 \\ -.015 \ .309 & -.014 \ .074 & .008 & 0.212 \end{pmatrix} \\ (x_o, y)^{K_{f_{io}}} &= (x_{o,1}x_{o,2}y, x_{o,1}x_{o,2}, x_{o,1}y, x_{o,1}, x_{o,2}y, x_{o,2})^T. \end{aligned}$$

REFERENCES

- Bartosiewicz, Z. (1988). Minimal polynomial realizations. *Math. Control Signals Systems*, 1, 227 – 237.
- Bazanella, A., Gevers, M., Coutinho, D., and Rui, R. (2014). Identifiability and excitability of a class of rational systems. In *Proceedings of the 53rd IEEE Conference on Decision and Control*, 6522 – 6527. IEEE, IEEE Press, New York.
- Becker, T. and Weispfenning, V. (1993a). *Gröbner bases*. Springer, Berlin.
- Becker, T. and Weispfenning, V. (1993b). *Gröbner bases: A computational approach to commutative algebra*. Graduate Texts in Mathematics. Springer, Berlin.
- Boulier, F. and Lemaire, F. (2009). Differential algebra and QSSA methods in biochemistry. In M. Basseville and A. Ticino (eds.), *Proceedings 15th IFAC Symposium System Identification (SYSID.2009)*, 33–38. IFAC, Elsevier, Amsterdam.
- Cox, D., Little, J., and O’Shea, D. (1992). *Ideals, varieties, algorithms: An introduction to computational algebraic geometry and commutative algebra*. Undergraduate Texts in Mathematics. Springer, Berlin.
- Fliess, M. (1990). Automatique en temps discret et algèbre aux diff’érences. *Forum Mathematicum*, 2, 213–232. URL <http://eudml.org/doc/141640>.
- Gevers, M., Bazanella, A., Coutinho, D., and Dasgoupta, S. (2013). Identifiability and excitability of polynomial systems (1). In *Proceedings of the 52nd IEEE Conference on Decision and Control*. IEEE, IEEE Press, New York.
- Golub, G. and Loan, C.V. (1983). *Matrix computations*. The Johns Hopkins University Press, Baltimore.
- Isidori, A. (1973). Direct construction of minimal bilinear realizations from nonlinear input-output maps. *IEEE Trans. Automatic Control*, 18, 626–631.
- Moor, B.D. (2014). Back to the roots: Solving polynomial systems with numerical linear algebra tools. In *Proc. 21st International Symposium on the Mathematical Theory of Networks and Systems (MTNS.2014)*. University of Groningen, Groningen.
- Müller-Quade, J. and Steinwandt, R. (2000). Gröbner basis applied to finitely generated field extensions. *J. Symbolic Computation*, 30, 469–490.
- Nemcova, J. and van Schuppen, J.H. (2009). Tutorial on system identification of polynomial and of rational systems. In M. Basseville and A. Ticino (eds.), *Proceedings 15th IFAC Symposium System Identification (SYSID.2009)*, 21–26. IFAC, Elsevier, Amsterdam.

- Němcová, J. (2010). Structural identifiability of polynomial and rational systems. *Math. Biosci.*, 223, 83–96.
- Němcová, J. and van Schuppen, J. (2010). Realization theory for rational systems: Minimal realizations. *Acta Applicandae Mathematicae*, 110, 605–626.
- Němcová, J., Petreczky, M., and van Schuppen, J.H. (2012). System reduction and identification of polynomial and of rational systems. In J. Schoukens (ed.), *Preprints of the 16th IFAC Symposium on System Identification*, 953–958. IFAC, IFAC, Laxenburg.
- Němcová, J. and van Schuppen, J.H. (2009). Realization theory for rational systems: The existence of rational realizations. *SIAM J. Control & Opt.*, 48, 2840 – 2856.
- Sontag, E. (1979). *Polynomial response maps*, volume 13 of *Lecture Notes in Control and Information Sciences*. Springer-Verlag, Berlin.
- van Overschee, P. and De Moor, B. (1996). *Subspace identification for linear systems*. Kluwer Academic Publishers, Dordrecht.
- Vidal, R. (2008). Recursive identification of switched ARX systems. *Automatica*, 44, 2274–2287.
- Zariski, O. and Samuel, P. (1958). *Commutative algebra – Volume 1*. Springer, New York.

Appendix A. NOTATION

The set of the integers is denoted by \mathbb{Z} and the positive integers by $\mathbb{Z}_+ = \{1, 2, \dots\}$. The set of the natural numbers is denoted by $\mathbb{N} = \{0, 1, 2, \dots\}$ and by \mathbb{N}^n its n -tuples. For any $n \in \mathbb{Z}_+$, denote $\mathbb{Z}_n = \{1, 2, \dots, n\}$. The set of the real numbers is denoted by \mathbb{R} , the positive real numbers by $\mathbb{R}_+ = [0, \infty)$, and the strictly-positive real numbers by $\mathbb{R}_{s+} = (0, \infty)$. The vector space of n -tuples of the real numbers is denoted by \mathbb{R}^n , for $n \in \mathbb{Z}_+$. The set of matrices with entries in the real numbers of size $n \times m$, for $n, m \in \mathbb{Z}_+$, is denoted by $\mathbb{R}^{n \times m}$. A diagonal matrix of the set of square real matrices $\mathbb{R}^{n \times n}$ for $n \in \mathbb{Z}_+$ is a matrix such that its off-diagonal elements all zero and the set of such matrices is denoted by $\mathbb{R}_{\text{diag}}^{n \times n}$. The subset of positive diagonal matrices is defined by the condition that $D_{i,i} \geq 0$ for all $i \in \mathbb{Z}_n$ and it is denoted by $\mathbb{R}_{\text{diag},+}^{n \times n}$. Similarly, $\mathbb{R}_{\text{diag},s+}^{n \times n}$.

Definition A.1. The *truncation operation* of a positive diagonal matrix based on the l_1 -norm of the diagonal.

Data. $(n, D, r) \in (\mathbb{Z}_+ \times \mathbb{R}_{\text{diag},+}^{n \times n} \times (0, 1))$, $D \neq 0$, where r is an approximation threshold. Assume that $D_{1,1} \geq D_{2,2} \geq \dots \geq D_{n,n} \geq 0$.

- (1) Compute the l_1 -norm of the diagonal elements of the diagonal matrix D , $\|\text{diag}(D)\|_{l_1} = \sum_{i=1}^n D_{i,i}$.
- (2) Compute $n_r = \text{argmin}_{j \in \mathbb{Z}_n} \{\sum_{i=1}^j D_{i,i} / \|\text{diag}(D)\|_{l_1} \geq r\}$.
- (3) Construct the approximant positive diagonal matrix

$$\begin{aligned} D_{n_r, i, i} &= D_{i, i}, \forall i \in \{1, 2, \dots, n_r\}, \quad D_{n_r} \in \mathbb{R}_{\text{diag},+}^{n_r \times n_r}, \\ D_r &= \begin{pmatrix} D_{n_r} & 0 \\ 0 & 0 \end{pmatrix} \in \mathbb{R}_{\text{diag},+}^{n \times n}. \end{aligned} \quad (\text{A.1})$$

- (4) Output $(n_r, D_r, \text{table}_1) \in (\mathbb{N} \times \mathbb{R}_{\text{diag},+}^{n \times n} \times \mathbb{R}^{n \times 2})$, where $\text{table}_1 = \{(j, \sum_{i=1}^j D_{i,i} / \|\text{diag}(D)\|_{l_1}), j \in \mathbb{Z}_n\}$.

Appendix B. MONOMIALS AND MONOMIAL VECTORS

Definition B.1. Consider a set of $n \in \mathbb{Z}_+$ commutative variables denoted by $x = (x_1, \dots, x_n)$. A *monomial* is a term of a polynomial defined by the formulas,

$$x^k = \prod_{i=1}^n x_i^{k(i)} = x_1^{k(1)} x_2^{k(2)} \dots x_n^{k(n)}, \quad k \in \mathbb{N}^n,$$

$$G_{\text{mon}}[x] = \{x^k \mid \forall k \in \mathbb{N}^n\}, \quad A(G_{\text{mon}})[x] = A_{\mathbb{R}^n}[x],$$

$$p(x) = \sum_{k \in \mathbb{N}^n} c(k)x^k \in \mathbb{R}[x], \quad \forall k \in \mathbb{N}^n, \quad c(k) \in \mathbb{R}.$$

Call x^k a *monomial* in the indeterminates x , a vector $k \in \mathbb{N}^n$ a *power vector*, \mathbb{N}^n a *set of power vectors*, $G_{\text{mon}}[x]$ the set of all monomials of x , and $p \in \mathbb{R}[x]$ a *polynomial in monomial representation*.

There exists a bijective correspondence between the set of power vectors \mathbb{N}^n and the set $G_{\text{mon}}[x]$ of monomials in the indeterminates x , described by the map $k \mapsto x^k$. The partially-ordered set \mathbb{N}^n may be equipped with a monomial ordering. Below the specific monomial ordering called the *lexicographic order relation* on the index set of power vectors \mathbb{N}^n will be used, see (Cox et al., 1992, Def. 2.2.3). It is denoted by $>_{\text{lex}}$. By the bijective correspondence between \mathbb{N}^n and $G_{\text{mon}}[x]$ the lexicographic ordering of \mathbb{N}^n is transformed into a lexicographic ordering on $G_{\text{mon}}[x]$ which is again denoted by $>_{\text{lex}}$ and which will be called the *lexicographic ordering* of $G_{\text{mon}}[x]$.

Definition B.2. Define the *power-bounded monomial vector* of a set of $n \in \mathbb{Z}_+$ commutative variables by the following formulas.

$$k_{\text{max}} \in \mathbb{N}^n,$$

$$\mathbb{N}^n(k_{\text{max}}) = \{k \in \mathbb{N}^n \mid 0 \leq k(i) \leq k_{\text{max}}(i), \quad \forall i \in \mathbb{Z}_n\},$$

$$k^* = |\mathbb{N}^n(k_{\text{max}})| = \prod_{i=1}^n (k_{\text{max}}(i) + 1) \in \mathbb{Z}_+,$$

$$\text{choose } d_v \in \mathbb{Z}_+, \quad 0 < d_v \leq k^*,$$

$$\mathbb{N}^{d_v \times n}(k_{\text{max}}) = \{K \in \mathbb{N}^{d_v \times n} \mid 0 \leq K(i, j) \leq k_{\text{max}}(j)\},$$

$$\text{choose } K_v \in \mathbb{N}^{d_v \times n}(k_{\text{max}}), \text{ and define,}$$

$$v = v(x, n, K_v) = x^{K_v} \in \mathbb{R}^{d_v}, \quad (\text{B.1})$$

$$v(x, n, K_v)_i = x_1^{K_v(i,1)} x_2^{K_v(i,2)} \dots x_n^{K_v(i,n)}; \quad h \in \mathbb{R}^{d_v},$$

$$p(x) = \sum_{k \in K_v} c(k)x^k = h^T v(x, n, K_v) = h^T x^{K_v}.$$

Call k_{max} the *maximal power vector* and $k_{\text{max}}(i)$ the *maximal power* of x_i ; $\mathbb{N}^n(k_{\text{max}})$ the *bounded power vector set*, which set inherits the lexicographic ordering of the elements of \mathbb{N}^n ; $K_v \in \mathbb{N}^{d_v \times n}(k_{\text{max}})$ the *power matrix* of the monomial vector $v(x, n, K_v)$, where the row elements of K_v are the powers vectors of the vector v in decreasing lexicographic order; $v(x, n, K_v)$ a (symbolic) *monomial vector*, which contains all monomials indexed by $K_v \in \mathbb{N}^{d_v \times n}(k_{\text{max}})$ in their lexicographic order from high to low order; the number of components of this monomial vector equals $d_v \leq k^*$; and finally call the equation $p(x) = h^T v(x, n, K_v) = h^T x^{K_v}$, the *power-bounded monomial representation* of the polynomial p .

Example B.3. There follows an example of a monomial vector.

$$x = (x_1, x_2), \quad n = 2, \quad k_{\text{max}} = (2, 1)^T,$$

$$d_v = k^* = (2 + 1) \times (1 + 1) = 6,$$

$$v(x, 2, (2, 1)) = \begin{pmatrix} x_1^2 x_2 \\ x_1^2 \\ x_1 x_2 \\ x_1 \\ x_2 \\ 1 \end{pmatrix}; \quad K_v = \begin{pmatrix} 2 & 1 \\ 2 & 0 \\ 1 & 1 \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix} \in \mathbb{N}^{6 \times 2} \left(\begin{pmatrix} 2 \\ 1 \end{pmatrix} \right).$$

Appendix C. POLYNOMIALS

In this appendix and the next one several aspects of the commutative algebra of polynomials maps are described including algebraic geometry. See Becker and Weispfenning (1993a); Moor (2014), Cox et al. (1992), and Zariski and Samuel (1958).

Let $n \in \mathbb{Z}_+$ be a positive integer. The *ring of polynomials* in n variables with real coefficients is denoted by $\mathbb{R}[x_1, \dots, x_n]$. The simplified notation of $\mathbb{R}[x]$ will be used if it is understood that the variable x has n components. Examples are $2x^2 + 3x + 4 \in \mathbb{R}[x]$ and $21x_1^2 x_2 + 11x_1 x_2 + 1x_2 \in \mathbb{R}[x_1, x_2]$.

Below polynomial functions of tuples of the real numbers $X = \mathbb{R}^n$ are needed. A *polynomial function* on \mathbb{R}^n is a map $p : X \rightarrow \mathbb{R}^n$ for which there exists a set of polynomials $q_1, \dots, q_n \in \mathbb{R}[x_1, x_2, \dots, x_n]$ such that $p_i = q_i$ on \mathbb{R}^n for all $i \in \mathbb{Z}_n$. Denote by A_X the set of all polynomials on $X = \mathbb{R}^n$.

Definition C.1. The *monomial representation* of a finite set of polynomials in the indeterminates $x = x_1 \dots x_{d_x}$ power bounded by the vector $k_{\text{max}} \in \mathbb{N}^n$, where the polynomials are the components of Lx^K , is defined by the notation,

$$G = \left\{ Lx^K \in \mathbb{R}[x] \mid (L, K) \in (\mathbb{R}^{\text{card}_G \times d_v} \times \mathbb{N}^{d_v \times d_x}(k_{\text{max}})) \right\}. \quad (\text{C.1})$$

Example C.2. Consider the set of polynomials,

$$L = \begin{pmatrix} 0.1 & 0.2 \\ 0.3 & 0.4 \end{pmatrix}, \quad K = \begin{pmatrix} 3 & 1 \\ 1 & 2 \end{pmatrix}, \quad x^K = \begin{pmatrix} x_1^3 x_2 \\ x_1 x_2^2 \end{pmatrix},$$

$$\left\{ p(x) = Lx^K = \begin{pmatrix} 0.1x_1^3 x_2 + 0.2x_1 x_2^2 \\ 0.3x_1^3 x_2 + 0.4x_1 x_2^2 \end{pmatrix} \right\}.$$

Appendix D. A SET OF GENERATORS

A *subalgebra* A_1 of the algebra A_X is a subset $A_1 \subseteq A_X$ such that the algebraic operations of A_1 are obtained from those of A_X by restriction and such that it is also an algebra itself in terms of those operations.

Consider a subset $G \subseteq A_X$. The smallest subalgebra of A_X containing G exists, it is called the *algebra generated* by the set G , and it is denoted by $A_X(G) \subseteq A_X$. A subalgebra $A_1 \subseteq A_X$ is called *finitely generated* if there exists a finite subset $G_f \subset A_X$ such that $A_1 = A_X(G_f)$.

Definition D.1. Call the finite set G , see (C.1), a *generator set* of the subalgebra $A \subseteq \mathbb{R}[x]$ and any row component of $Lx^K \in G$ a *generator* of A if $A = A(G)$. Call it a *nontrivial generator set* if no column of the matrix L in the representation is entirely zero. Call G a *minimal generator set* of the algebra $A = A(G)$ if it is nontrivial and for any other generator set H it holds that $\text{card}_G \leq \text{card}_H$. A *set of generators* of a *finite set* of polynomials H is a finite set of polynomials G such that

$$A(H) = A(G), \text{ where, } H = \{p_1, \dots, p_{\text{card}_H} \in \mathbb{R}[x]\}, \\ G = \{g_1, \dots, g_{\text{card}_G} \in \mathbb{R}[x]\}.$$

The set of real numbers is also a ring. A finite subset $\{p_1, \dots, p_k\} \subset \mathbb{R}[x]$ is called *algebraically dependent* over \mathbb{R} if there exists a nonzero polynomial $f \in \mathbb{R}[p]$ such that $f(p_1, \dots, p_k) = 0$. It is called *transcendental* otherwise. See (Zariski and Samuel, 1958, I, §17, p.28). An extension not described here because of lack of space is to define a transcendence basis for the algebraic structure used which then allows the use of the algorithms of Müller-Quade and Steinwandt (2000) for the computation of such a basis.

Problem D.2. Consider a finite set of polynomials $H \subset \mathbb{R}[x]$. Construct a minimal set of generators $G \subset \mathbb{R}[x]$ of H .

The above problem is *equivalent* to the problem of constructing a polynomial factorization of a polynomial map as defined next.

Definition D.3. A *polynomial factorization* of a polynomial map $y = f(u)$ with $d_y, d_u \in \mathbb{Z}_+, f : \mathbb{R}^{d_u} \rightarrow \mathbb{R}^{d_y}$, is defined to be a factorization of the form,

$$y = f(u) = h(g(u)) = h(x), \quad h \in \mathbb{R}[x], \quad (\text{D.1})$$

$$x = g(u) \in X \subseteq \mathbb{R}^{d_x}, \quad g \in \mathbb{R}[u], \quad (\text{D.2})$$

$$G_f = \{f_1, \dots, f_{d_y} \in \mathbb{R}[u]\}, \quad G_g = \{g_1, \dots, g_{d_x} \in \mathbb{R}[u]\}, \\ A(G_f) = A(G_g), \quad (\text{D.3})$$

hence G_g is a set of generators of $A(G_f)$. Note that G_g is a minimal set of generators if $d_x \in \mathbb{Z}_+$ is minimal over all factorizations.

Appendix E. APPROXIMATION OF A GENERATOR SET

Problem E.1. The problem of *approximate polynomial factorization*. Consider a polynomial function $y = f(u)$ as defined in Section C. (The notation of this section differs from the main body of the paper.) Determine an approximate polynomial factorization of the form,

$$y = f(u) \approx h(g(u)) = h(x), \quad x = g(u), \quad (\text{E.1})$$

$$d_x \in \mathbb{Z}_+, \quad g \in \mathbb{R}[u], \quad h \in \mathbb{R}[x].$$

The approximation criterion of the expression $[f(u) - h(g(u))]$ is not specified.

Definition E.2. The *approximate polynomial factorization algorithm*. Data. $y = f(u)$, $d_y, d_u \in \mathbb{Z}_+, f \in \mathbb{R}[u]$.

- (1) Construct the approximation consisting of a linear-polynomial factorization by the algorithm of Def. E.4,

$$y = f(u) \approx C_r g_r(u) = C_r x_r, \quad x_r = g_r(u), \quad (\text{E.2})$$

$$d_{x_r} \in \mathbb{Z}_+, \quad x_r \in \mathbb{R}^{d_{x_r}}, \quad C_r \in \mathbb{R}^{d_y \times d_{x_r}}. \quad (\text{E.3})$$

- (2) For the linear-polynomial factorization of Step 1 construct a new generator set of, possibly, lower cardinality than before, or, equivalently, a polynomial factorization,

$$y = f(u) \approx C_r g_r(u) = h(g(u)) = h(x), \quad (\text{E.4})$$

$$x = g(u), \quad (\text{E.5})$$

$$d_x \in \mathbb{Z}_+, \quad x \in \mathbb{R}^{d_x}, \quad h \in \mathbb{R}[x], \quad g \in \mathbb{R}[u],$$

by polynomial factorizations of the components of g_r . For example, if for polynomial $g_{r,m}$ there exists a factorization of the form $g_{r,m} = g_{r,i} g_{r,j} + g_{r,s}$, where $g_{r,i}, g_{r,j}, g_{r,s}$

have lower powers than those of $g_{r,m}$, see (Becker and Weispfenning, 1993b, Sec. 5.1.).

Comments follow. (1) A Gröbner basis algorithm is not appropriate for Problem E.1 of the polynomial function $y = f(u)$ because that would not be an approximation. In addition, the computational complexity is too high. (2) A Gröbner basis algorithm may well be useful for Step 2 of Algorithm E.2. For the computations a simple procedure is used for Step 2 of Algorithm E.2. The procedure is not described here in detail because of lack of space.

The transformation for a polynomial factorization is briefly described. Note that if

$$y = f(u) \approx C_r g_r(u), \quad x = g_r(u), \quad (\text{E.6})$$

$$g_{r,m}(u) = x_{r,m} = x_{r,i} x_{r,j} = g_{r,i}(u) g_{r,j}(u), \text{ then,} \quad (\text{E.7})$$

$$x_r = (x_1 \dots x_{m-1} x_{m+1} \dots x_{d_x})^T,$$

$$x = (x_1 \dots x_{m-1} x_i x_j x_{m+1} \dots x_{d_x})^T,$$

$$= P_x (x_i x_j x_1 \dots x_{d_x})^T = P_x v(x_r) = P_x x_r^{K_{x_r}},$$

$$y \approx C_r x = C_r P_x v(x_r) = C v(x_r) = h(x_r), \quad (\text{E.8})$$

$$g(u) = P_{g_r} g_r(u), \quad (\text{E.9})$$

$$y \approx C_r x = C_r g_r(u) = h(x_r), \quad x_r = g(u), \quad (\text{E.10})$$

It is not claimed that the above procedure determines a minimal generator set which is not true in general.

Definition E.3. The *monomial equation of data matrices*. Consider the polynomial equation $y = f(u)$ and its monomial representation. Consider the case in which one is provided several tuples of values of input and output vectors, $\{(\bar{y}_i, \bar{u}_i) \in Y \times U \mid i = 1, 2, \dots, s\}$. A *monomial equation of the data matrices* for this set of tuples is then a linear map represented by the coefficient matrix H .

$$V_y = H V_u, \quad H \in \mathbb{R}^{d_{v_y} \times d_{v_u}}, \quad (\text{E.11})$$

$$V_y = (v(\bar{y}_1, d_y, K_{v_y}) \dots v(\bar{y}_s, d_y, K_{v_y})) \in \mathbb{R}^{d_{v_y} \times s},$$

$$V_u = (v(\bar{u}_1, d_u, K_{v_u}) \dots v(\bar{u}_s, d_u, K_{v_u})) \in \mathbb{R}^{d_{v_u} \times s}.$$

Definition E.4. *Linear approximation of a polynomial map*. (Golub and Loan, 1983, Sec. 6.1).

This algorithm is called *SVDtruncation* in Steps 5 and 11 of Algorithm 4.1. Data. $(d_y, d_u, d_{v_y}, d_{v_u}, s, V_y, V_u, r) \in (\mathbb{Z}_+^5 \times \mathbb{R}^{d_{v_y} \times s} \times \mathbb{R}^{d_{v_u} \times s} \times (0, 1))$.

- (1) Compute the singular value decomposition of the data matrix of the inputs,

$$V_u = V_1^T S V_2 \in \mathbb{R}^{d_{v_u} \times s}, \quad (\text{E.12})$$

$$V_1 \in \mathbb{R}^{d_{v_u} \times d_{v_u}}, \quad V_2 \in \mathbb{R}^{s \times s}, \text{ orthogonal matrices,}$$

$$S = \begin{pmatrix} D & 0 \\ 0 & 0 \end{pmatrix} \in \mathbb{R}^{d_{v_u} \times s}, \quad n_1 \in \mathbb{N},$$

$$D = \text{diag}(d_1, d_2, \dots, d_{n_1}) \in \mathbb{R}_{\text{diag}, s+}^{n_1 \times n_1},$$

$$d_1 \geq d_2 \geq \dots \geq d_{n_1} > 0.$$

- (2) Compute according to the algorithm of Def. A.1, of the diagonal matrix D its truncation D_n upto the approximation fraction,

$$(n, D_n, \text{table}_1) = \text{mdtrunc}(n_1, D), \quad (\text{E.13})$$

$$S_n^+ = \begin{pmatrix} D_n^{-1} & 0 \\ 0 & 0 \end{pmatrix} \in \mathbb{R}^{s \times d_{v_u}}, \quad D_n \in \mathbb{R}_{\text{diag}, s+}^{n \times n},$$

$$V_u^+ = V_2^T S_n^+ V_1 \in \mathbb{R}^{s \times d_{v_u}}, \quad H^* = V_y V_u^+ \in \mathbb{R}^{d_{v_y} \times d_{v_u}}.$$

(3) Compute the factorization according to,

$$H^* = [V_y V_2^T \begin{pmatrix} D_n^{-1} \\ 0 \end{pmatrix}] [(I_n \ 0) V_1] = CL, \quad (\text{E.14})$$

$$L = (I_n \ 0) V_1 \in \mathbb{R}^{n \times d_{v_u}}, \quad X = LV_u \in \mathbb{R}^{n \times s},$$

$$C = V_y V_2^T \begin{pmatrix} D_n^{-1} \\ 0 \end{pmatrix} \in \mathbb{R}^{d_{v_y} \times n}, \quad (\text{E.15})$$

$$V_y \approx H^* V_u = CLV_u = CX. \quad (\text{E.16})$$

(4) Output $(n, D_n, C, L, X, H^*, \text{table})$ with

$$\text{table} = \{(j, \sum_{i=1}^j D_{i,i} / \|\text{diag}(D)\|_{l_1}), j \in \mathbb{Z}_{n_1}\}.$$